

# Update on Machine Learning Studies at FAST

---

Auralee Edelen

FAST/IOTA Collaboration Meeting

6 June, 2017

# Temperature/Resonant Frequency Control for the RF Gun

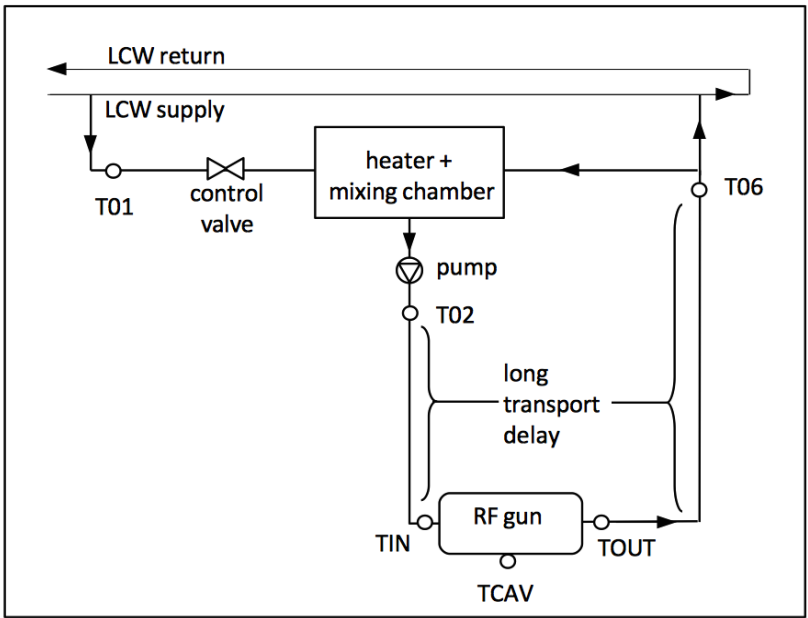
Resonant frequency controlled via gun temperature

PID control is undesirable in this case:

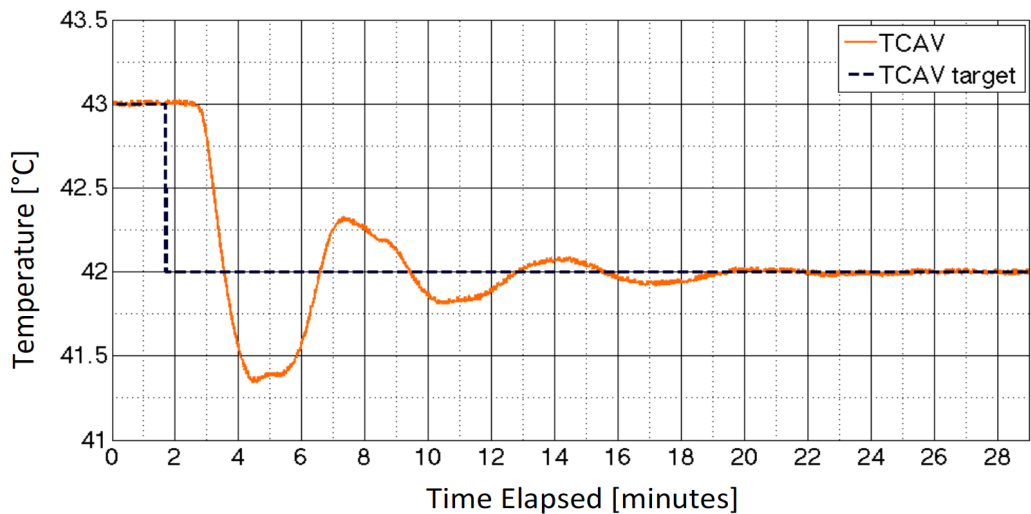
- Long transport delays and thermal responses
- Recirculation leads to secondary impact of disturbances
- Two controllable variables: heater power + valve aperture

Applied **model predictive control (MPC)** with a **neural network model** trained on measured data: **~ 5x faster settling time + no large overshoot**

Gun Water System Layout

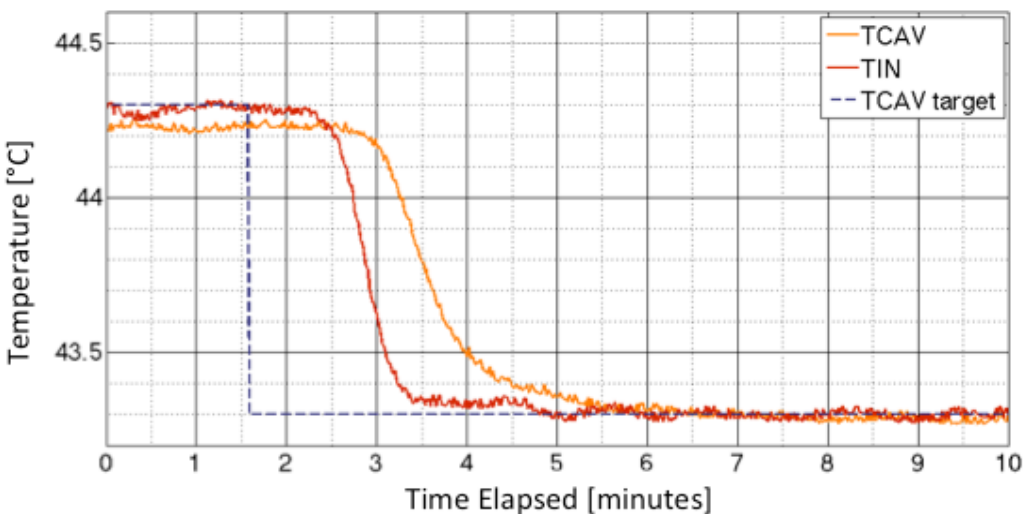


Existing Feedforward/PID Controller



*Note that the oscillations are largely due to the transport delays and water recirculation, rather than PID gains*

Model Predictive Controller



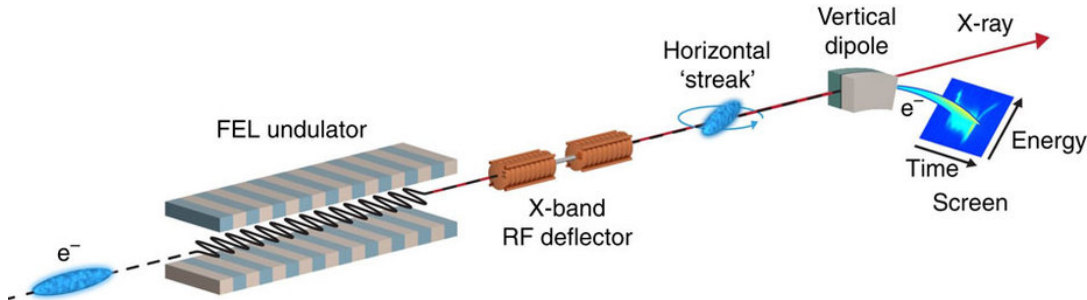
*More info: A. L. Edelen et al., IEEE TNS, vol. 63, no. 2, 2016*

*That was an example of using a learned model and predictive control for a system with long-term time dependencies...*

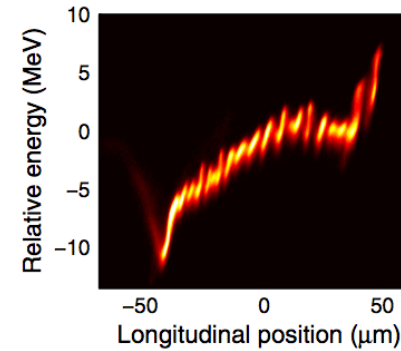
*... what are some areas where a learned control policy might be useful?*

- **Image diagnostics** → would be nice to use directly, and some yield relatively complicated information

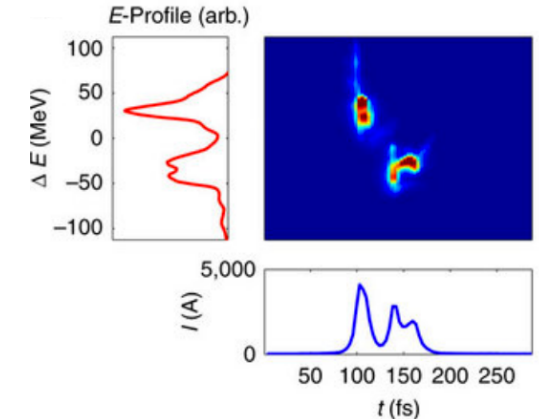
e.g. XTCAV at SLAC



*C. Behrens, et al., Nat. Commun. 5, 3762 (2014)*

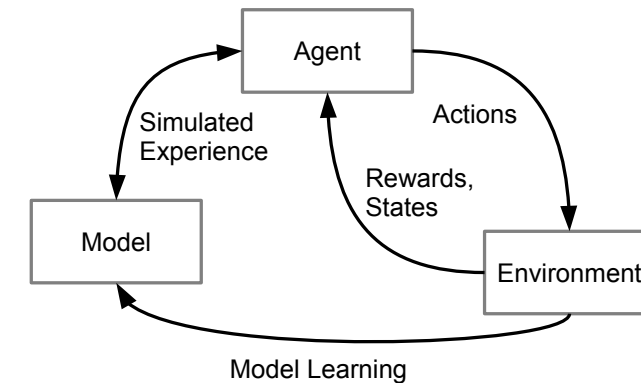


*D. Ratner, et al., PRSTAB 18, 030704 (2015)*



*A. Marinelli, et al., Nat. Commun. 6, 6369 (2015)*

- **Convolutional Neural Networks (CNNs)** → very good at image processing
- **Reinforcement Learning (RL)** → can learn control policies from data



***Why not try using image based diagnostics directly in learned control policies?***

***What's a relatively simple test case to start with?***

# Initial Study: Choose Gun Parameters Based on Laser Spot

## Motivation:

- Gun phase and solenoid strength tuned daily
- Asymmetries in initial laser distribution result in emittance asymmetries downstream
- Would be nice to obtain optimal gun phase and solenoid strength for a given initial laser distribution automatically (and perhaps prioritize x or y emittance to minimize)



*Example virtual cathode image  
(10 Aug. 2016)*

## Other perks:

- PARMELA simulation based on survey data already in existence (J. Edelen)
- Try out creating a fast NN modeling tool from slower-executing simulations

# Initial Study: Choose Gun Parameters based on Laser Spot

## Motivation:

- Gun phase and solenoid current

- Acceleration

- Velocity

- Spot

- di

- x

## Other

- PAR

- Try c

Why not just use online optimization?

Why not just fit a Gaussian to the laser spot to get the information instead of using images directly?

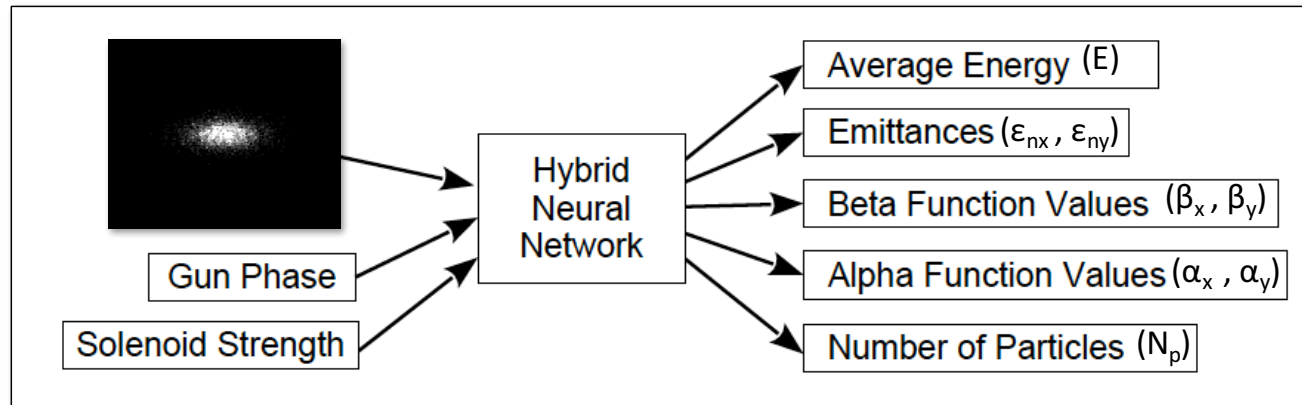
**The point of this study: explore this approach on a simple system (it's a stepping stone)**

- PAR already in existence (J. Edelen)

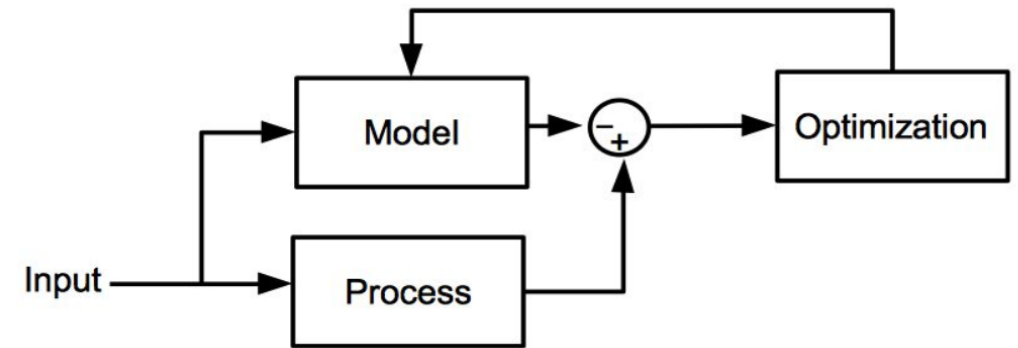
- Try c modeling tool from slower-executing simulations

# Initial Study: Steps

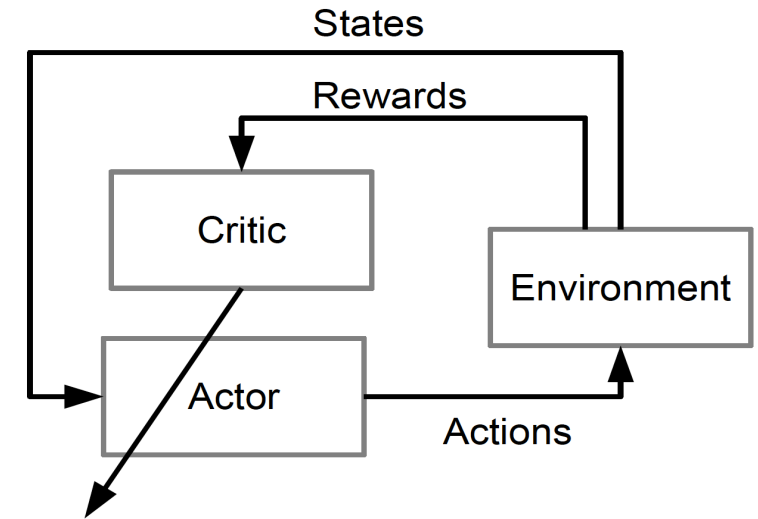
- Gather simulation data from PARMELA scans
- Create a NN model
  - Be certain that the necessary information can be extracted from the image, gun phase, and solenoid strength
- Train a RL controller using that model
- Extension beyond simulation (tentative):
  - Incorporate measured data into model and update controller
  - Carefully test on machine



*model inputs and outputs*



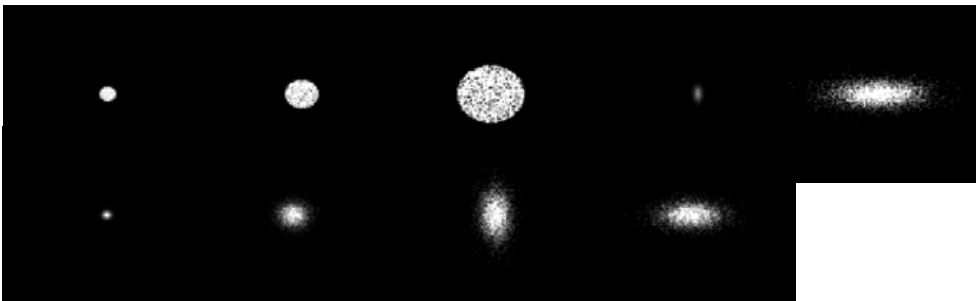
***Model Learning***



***Policy Learning***

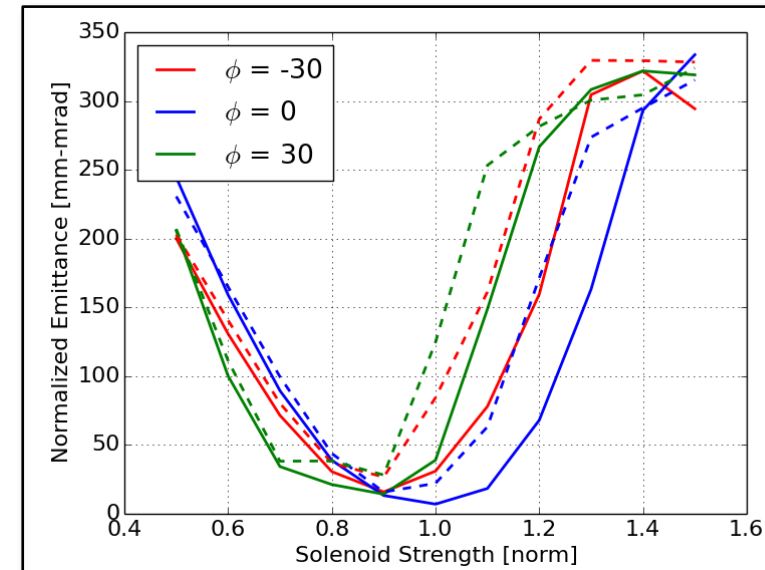
# CNN Model: Simulation Data

- PARMELA simulations from the gun up to the exit of CC2
  - 2-D space charge routine
  - Scanned gun phase, solenoid strength, initial beam distribution
- Two sets of data:
  - Fine scans (steps of  $5^\circ$  phase, 5% sol. str.) for sims just past the gun
  - Coarse scans (steps of  $10^\circ$  phase, 10% sol. str.) for sims up through CC2
- Simulated “virtual cathode images”
  - Going from VCI  $\rightarrow$  initial beam distribution ok from prior work
  - Initial beam distribution  $\rightarrow$  simulated VCI probably ok
  - Obviously very “well-behaved” examples



Parameter Ranges used for Model Training

| Parameter               | Gun Data  |           | CC2 Data  |           |
|-------------------------|-----------|-----------|-----------|-----------|
|                         | Max Value | Min Value | Max Value | Min Value |
| $N_p$                   | 5001      | 1015      | 5001      | 1004      |
| $\epsilon_{nx}$ [m-rad] | 2.50E-04  | 1.60E-06  | 4.00E-04  | 9.10E-07  |
| $\epsilon_{ny}$ [m-rad] | 2.40E-04  | 1.60E-06  | 4.00E-04  | 8.50E-07  |
| $\alpha_x$ [rad]        | 14.1      | -775.1    | 0.8       | -149.8    |
| $\alpha_y$ [rad]        | 14.5      | -797      | 0.7       | -154.5    |
| $\beta_x$ [m/rad]       | 950.4     | 7.90E-02  | 820.2     | 0.7       |
| $\beta_y$ [m/rad]       | 896.8     | 8.40E-02  | 845.7     | 0.81      |
| E [MeV]                 | 4.6       | 3.2       | 47.2      | 42.8      |



Simulation predictions after CC2. Dashed lines are x-emittance, solid lines are y-emittance. Caveat: doesn't take into account coupling...later changed NN setup to predict sigma matrix, and also used a 3D space charge routine.

For normalized sol strength, 1 is the setting that produces a peak axial field of 1.8 kG



# CNN Model: Performance

| Parameter       | Train. MAE | Train. STD | Val. MAE | Val. STD |
|-----------------|------------|------------|----------|----------|
| $N_p$           | 69.5       | 79.8       | 70.7     | 75.7     |
| $\epsilon_{nx}$ | 2.30E-06   | 3.50E-06   | 2.40E-06 | 3.20E-06 |
| $\epsilon_{ny}$ | 2.30E-06   | 3.40E-06   | 2.40E-06 | 3.20E-06 |
| $\alpha_x$      | 9          | 14.9       | 10.9     | 16       |
| $\alpha_y$      | 8.8        | 15.3       | 10.8     | 16.1     |
| $\beta_x$       | 12.1       | 17.6       | 14.8     | 18.9     |
| $\beta_y$       | 11.7       | 16.7       | 14.3     | 17.9     |
| E               | 4.90E-03   | 4.90E-03   | 5.50E-03 | 6.00E-03 |

*Performance for the predictions after the gun*

| Parameter       | Train. MAE | Train. STD | Val. MAE | Val. STD |
|-----------------|------------|------------|----------|----------|
| $N_p$           | 103.7      | 141.2      | 123.3    | 176.8    |
| $\epsilon_{nx}$ | 1.00E-05   | 1.20E-05   | 1.20E-05 | 1.60E-05 |
| $\epsilon_{ny}$ | 1.00E-05   | 1.30E-05   | 1.20E-05 | 1.50E-05 |
| $\alpha_x$      | 3.4        | 6.6        | 3.1      | 5.9      |
| $\alpha_y$      | 3.4        | 6.6        | 3.1      | 5.9      |
| $\beta_x$       | 16.3       | 33.5       | 14.7     | 27.8     |
| $\beta_y$       | 16.4       | 33.6       | 14.8     | 27.5     |
| E               | 4.00E-02   | 3.90E-02   | 4.60E-02 | 6.20E-02 |

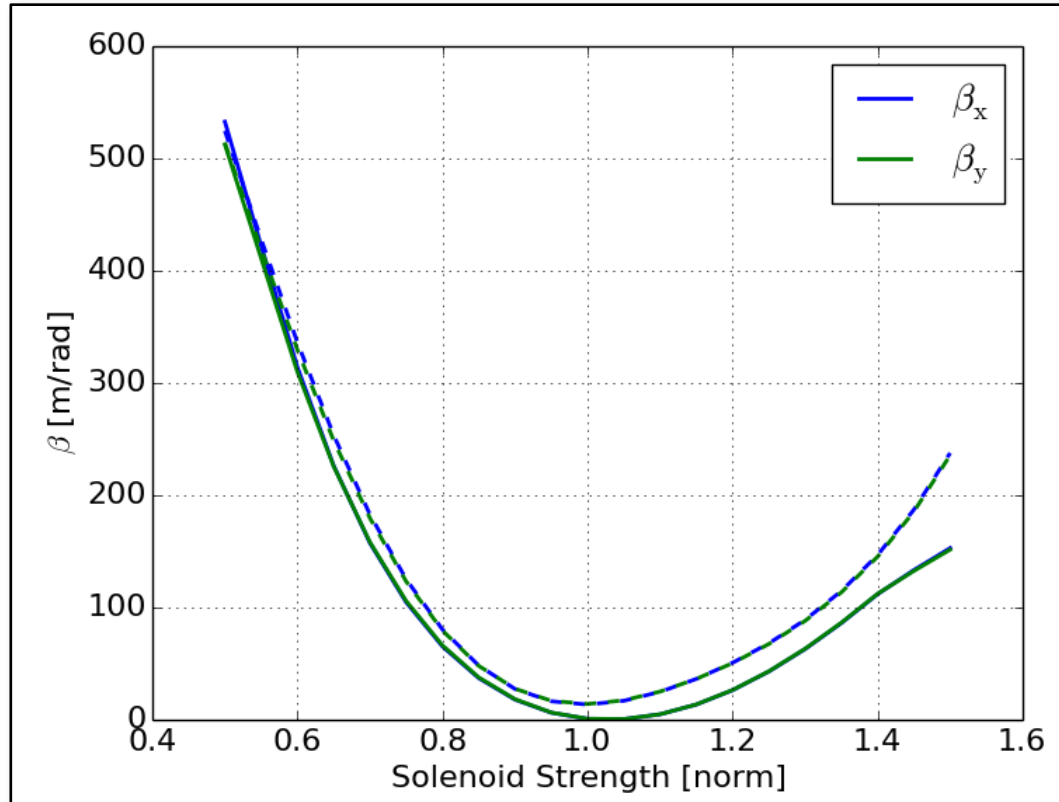
*Performance for the predictions after CC2*

For the gun data, all MAEs are between 0.4% and 1.8% of the parameter ranges.  
For the CC2 data, all MAEs are between 0.9% and 3.1% of the parameter ranges.

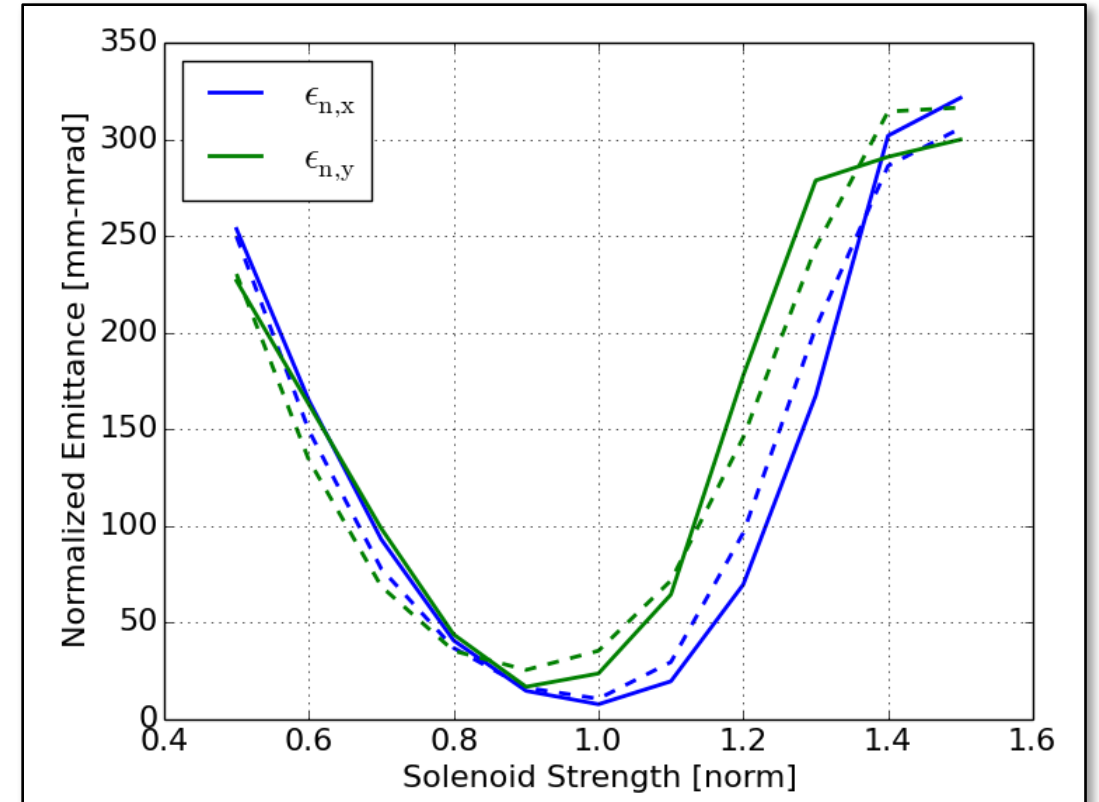
→ *Not bad for such a small training set*

# CNN Model: Two Representative Plots

Dashed lines are NN predictions and solid lines are simulation results



Top-hat initial beam, 0° RF phase, after gun



Asymmetric Gaussian initial beam, 0° RF phase, after CC2

## Overview

**Main Idea: make better use of image-based diagnostics using convolutional neural networks (CNNs), especially for direct incorporation into particle accelerator control systems.**

Here, we show the result of a **first step** toward this goal: our trained CNN can be used to **predict multiple simulated downstream beam parameters** at FAST using simulated virtual cathode laser images, gun phases, and solenoid strengths. This model is fast-executing, captures the dynamics of the simulation, and could already be used for fast optimization studies. With additional training on measured data, it could be used in model predictive controller.

## Beam Dynamics Simulations of the FAST Low Energy Beamline

Simulations of the FAST low energy beamline were conducted using PARMELA [5]. Included are the electron gun, both superconducting capture cavities, and the intermediate beam-line elements. A 2D space charge routine is also included. The field maps of the solenoid assembly, gun, and capture cavities used for the PARMELA simulations were generated using Poisson Superfish [6].

**Predicted Parameters:**  
—number of transmitted particles  
—transverse emittances  
—alpha and beta function values  
—average beam energy

**Two Sets of Data:**  
Fine scans for parameters after the gun  
Coarse scans for parameters after capture cavity two (CC2)

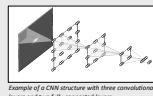
**Parameter Scans:**  
Gun phase:  $\pm 180^\circ$  in  $5^\circ$  and  $10^\circ$  steps  
Solenoid strength: 0.5 to 1.5 in 5% and 10% steps  
(1.0 represents the setting that produces a peak axial field of 3.8 kGauss)

**Initial Transverse Beam Distributions:**  
Fine scans: three top-hat distributions  
Coarse scans: three top-hat distributions and six Gaussian distributions



## Convolutional Neural Networks

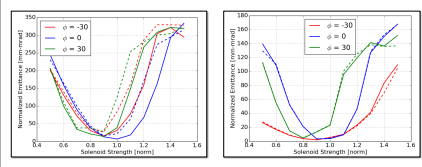
CNNs use **learned filters** to pick out **image features** that are relevant to the given training task. Recently, CNNs have yielded impressive results in the area of computer vision, especially for image recognition tasks [2].



They are also starting to be used in physics-related applications, such as automatic classification of galaxies [3] and neutrino events [4].

## Two Representative Examples of the Simulation Results

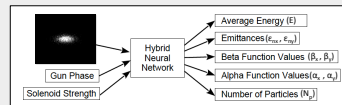
Here we show transverse emittances as a function of solenoid strength for three gun phases ( $\phi$ ). Dashed lines denote x emittance, and solid lines denote y emittance. The figure at left shows values after CC2 for a Gaussian initial beam distribution (x RMS width of 0.6 mm and a y RMS width of 1.2 mm). The figure at right shows values after the gun for a 0.6 mm width top-hat initial beam distribution.



## Neural Network Architecture

Instead of a classification task, here we use a CNN in a **regression task**. We also created a novel **hybrid structure** that joins a CNN and fully-connected NN to incorporate both image-based data and non-image-based data into the model. Initial beam distributions from the simulations were converted into images and used as simulated "virtual cathode images."

**Structure details:**  
• **3 convolutional layers:**  
16 5x5 filters, 16 3x3 filters, 10 3x3 filters  
• **3 fully-connected layers:**  
150, 70, and 8 nodes  
• Non-image data bypass the convolutional layers



## Training

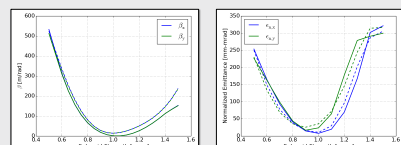
The network was trained using a combination of the ADDELTA [7] and Adam [8] optimization algorithms until performance on the validation data set began to deteriorate (i.e. training was stopped before significant overfitting occurred).

**Network Initialization:**  
Weights: layer-by-layer method described in [9]  
Biases: normal distribution (std. 0.01, mean 0)

**Training and Validation Sets:**  
Removed cases with <1000 transmitted particles  
Pseudo-random selection of validation data  
Final ranges of data sets shown at right  
Final number of samples:  
— Gun: 1395 for training, 200 for validation  
— CC2: 894 for training, 600 for validation

| Parameter             | Gun Data  |           | CC2 Data  |           |
|-----------------------|-----------|-----------|-----------|-----------|
|                       | Min Value | Max Value | Min Value | Max Value |
| $\phi$                | -180      | 180       | -180      | 180       |
| $N_p$                 | 2,500-04  | 3,400-06  | 4,000-08  | 9,100-07  |
| $\epsilon_{x,y}$ (mm) | 2,400-04  | 1,400-06  | 4,000-08  | 8,500-07  |
| $\beta_x$ (mm)        | 14.1      | 775.1     | 0.8       | 140.8     |
| $\beta_y$ (mm)        | 18.5      | 797       | 0.7       | 134.5     |
| $\alpha_x$ (mm)       | 990.8     | 7,900-02  | 820.2     | 0.7       |
| $\alpha_y$ (mm)       | 890.8     | 8,400-02  | 845.7     | 0.81      |
| $E$ (MeV)             | 4.5       | 1.2       | 47.2      | 40.8      |

## Performance In Predicting Beam Parameters after the Gun and the Second Capture Cavity



Transverse beta function values after the gun as a function of normalized solenoid strength. This is for a top-hat initial distribution and a gun RF phase of  $0^\circ$ . The dashed lines are NN predictions and the solid lines are simulated values.

| Parameter    | Train MAE | Train STD | Val MAE  | Val STD  |
|--------------|-----------|-----------|----------|----------|
| $\beta_x$    | 8.9       | 9.8       | 10.7     | 10.9     |
| $\beta_y$    | 2,100-06  | 3,500-06  | 2,400-06 | 3,200-06 |
| $\alpha_x$   | 2,100-06  | 3,400-06  | 2,400-06 | 3,200-06 |
| $\alpha_y$   | 9         | 14.3      | 10.9     | 10.9     |
| $N_p$        | 8.8       | 15.7      | 10.8     | 10.1     |
| $E$          | 12.1      | 17.6      | 14.8     | 18.9     |
| $\epsilon_x$ | 11.7      | 16.7      | 14.1     | 17.8     |
| $\epsilon_y$ | 4,900-03  | 4,900-03  | 5,500-03 | 6,000-03 |

Performance for the predictions after the gun.

The tables show the NN's performance in terms of mean absolute error (MAE) and standard deviation (STD) over the training and validation sets. The figures highlight two representative data sets to show how well the NN can predict the beam parameters.

Despite the small training set and the large number of predicted parameters, the neural network is able to capture the emittance asymmetry after CC2 that arises from the asymmetry of the initial laser distribution.

For the gun data, all MAEs are between 0.4% and 1.8% of the parameter ranges. For the CC2 data, all MAEs are between 0.9% and 3.1% of the parameter ranges.

## Conclusions and Next Steps

Despite the small number of samples in the training set, the **neural network performs well in predicting the downstream beam parameters** given solenoid strengths, gun phases, and simulated virtual cathode images. The neural network is able to capture the emittance asymmetry after CC2 that arises from the asymmetry of the initial laser distribution. This **fast-executing model** could already be used for quick optimization studies.

Presently, we are extending this work to include measured training data from the machine. For those studies, beam alignment will be used as an additional predicted parameter. Once the model is updated with measured data, we plan to train a neural network controller and test it on the machine.

This work was presented at NAPAC '16

- TUPOA51, <https://arxiv.org/abs/1612.05662>
- One of the student poster contest winners

# Present Status and Next Steps

- **Improving the quality of the setup:**

- Predicting the full sigma matrix
- More realistic initial distributions
- Using 3D space charge routine
- Using locally-connected layers
- Switching to ASTRA

( greater execution speed → more training data)

- **Next steps (in tandem):**

- Finish simulation study with present setup
- Extend to phase space manipulation simulation study
- Solidify plans for incorporating measured data and testing controller
  - Need to align available inputs/controllable variables (e.g. sigma matrix vs. info from emittance monitors, rotation of quads, etc.)
  - Also depends on run schedule, status of new emittance monitors, solid time with consistent setup, etc.

- **Expanding scope to phase space manipulations:**

- Specify a target sigma matrix
- Include quads after CC2, capture cavity phases, etc.
- Collaborating with NIU:
  - RTFB transform is a possible application
  - Alex Halavanau running simulation scans with NIU's newer model → more training data

*Also, if you have some other possible application and have or can easily obtain training data: don't hesitate to get in touch!*